



## A nonlinear minimax optimization program not requiring derivatives

Madsen, Karl; Schjær-Jacobsen, Hans

*Published in:*

I E E E Transactions on Antennas and Propagation

*Link to article, DOI:*

[10.1109/TAP.1977.1141575](https://doi.org/10.1109/TAP.1977.1141575)

*Publication date:*

1977

*Document Version*

Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*

Madsen, K., & Schjær-Jacobsen, H. (1977). A nonlinear minimax optimization program not requiring derivatives. *I E E E Transactions on Antennas and Propagation*, 25(3), 454-456. <https://doi.org/10.1109/TAP.1977.1141575>

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# A Nonlinear Minimax Optimization Program not Requiring Derivatives

**NAME:** MINI2W.  
**PURPOSE:** Suppose we have  $m$  functions  $f_j(\mathbf{x})$ ,  $j = 1, 2, \dots, m$  (which may be real or complex), and each is a function of  $n$  real parameters (or variables)  $x_i$ ,  $i = 1, 2, \dots, n$ ,  $n \leq m$ . Perhaps we desire a solution (that is, a combination of the parameters  $x_i$ ) such that all functions  $f_j$  will vanish. With the minimax approach, we seek instead a solution that minimizes the absolute magnitude of that function having the largest absolute magnitude. Thus, the program described minimizes the objective function  $F(\mathbf{x}) = \max_{1 \leq j \leq m} |f_j(\mathbf{x})|$ ,  $\mathbf{x} = (x_1, \dots, x_n)$ . It is *not* required that the user provide derivatives of the functions  $f_j$ . If some of these functions are complex, each complex function may be programmed as two real functions in order to use these subroutines. The relevance of the present program to design of antennas is established in [1] where nonuniformly spaced arrays were optimized to yield equal (minimum) sidelobes using uniform element excitation.

**LANGUAGE:** Fortran IV.  
**AUTHORS:** Karl Madsen, Institute for Numerical Analysis, The Technical University of Denmark, DK-2800 Lyngby, Denmark. Hans Schjær-Jacobsen was with the Electromagnetics Institute, The Technical University of Denmark, Lyngby, Denmark. He is now with the Institut für Hochfrequenztechnik, Technische Universität Braunschweig, D-3300, Braunschweig, Germany.

**AVAILABILITY:** ASIS-NAPS Document NAPS 02826.  
**DESCRIPTION:** The method is described in [1] and [2]. It is iterative and at each iteration the nonlinear problem under consideration is replaced by a linear approximation. The resulting linear minimax problem is solved subject to bounds on the step length by a method which is similar to the so-called exchange algorithm, [3].

MINI2W is a pilot subroutine, the aim of which is to simplify the work of the user. It calls 4 other subroutines, 3 of which are in the package, while the last one, for calculating function values, must be written by the user.

The call of MINI2W must have the following form:

CALL MINI2W (FUNC,N,M,X,DX,DXMAX,STEPL,EPS,MAXFUN,W,IW).

Manuscript received January 19, 1976.

See NAPS document 02826 for 10 pages of supplementary material. Order from ASIS/NAPS c/o Microfiche Publications, 440 Park Avenue South, New York, NY 10016. Remit in advance for each NAPS accession number. Make checks payable to Microfiche Publications. Photocopies are \$5.00. Microfiche are \$3.00. Outside of the United States and Canada, postage is \$2.00 for a photocopy or \$1.00 for a fiche.

## LIST OF PARAMETERS.

FUNC is the name of a subroutine written by the user. It must have the form

SUBROUTINE FUNC (N,M,X,B)

REAL\*8 X(N), B(M).

It must calculate the residual function values  $f_j(\mathbf{x})$  for the parameter values  $\mathbf{x} = (X(1), X(2), \dots, X(N))$  and store these in the following way  $B(j) = f_j(\mathbf{x})$ ,  $j = 1, \dots, M$ .

*Note:* The name of this user subroutine, which can be any name of the user's own choice, must appear in an EXTERNAL statement in the calling program.

**N** An INTEGER variable and must be set to  $n$ , the number of unknowns. Its value must be positive, and it is not changed by the subroutine.

**M** An INTEGER variable and must be set to  $m$ , the number of residual functions. Its value must be positive, and it is not changed by the subroutine.

**X** A REAL\*8 array which must be set by the user to an approximation to the solution,  $X(i) = x_i$ ,  $i = 1, \dots, N$ . On exit X will contain the best solution found by the subroutine.

**DX** is a REAL\*8 variable which controls the stop length of the iterative method used. It must be set by the user to an initial value corresponding to the starting vector  $\mathbf{x}^{(0)}$ . DX should be chosen so that in the region  $\{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}^{(0)}\| < DX\}$  the functions  $f_j$  can be approximated reasonably good by linear functions. If the functions are nearly linear DX could be set to an approximate value of the distance between the starting vector and the solution, but if more curvature is present this value may be too large. In general we suggest  $DX = 0.1 * \|\mathbf{x}^{(0)}\|$ . However, it is normally not severe to choose a bad initial value of DX, since DX is adjusted by the routine during the iteration. The value of DX must be positive.

**DXMAX** A REAL\*8 variable which must be set by the user to a general upper bound of the step length. DXMAX may be set to an estimate of the distance between the starting vector and the solution vector. Its value must be positive.

**STEPL** A REAL\*8 variable which must be set by the user. It is used by the subroutine to calculate an initial approximation to the derivative matrix, which is found by the differences

$$\frac{\partial f_j}{\partial x_i} \approx \{f_j(x_1^{(0)}, \dots, x_i^{(0)} + \text{STEPL}, \dots, x_n^{(0)}) - f_j(\mathbf{x}^{(0)})\} / \text{STEPL}$$

where  $\mathbf{x}^{(0)}$  is the starting vector. A value near  $0.001 * \|\mathbf{x}^{(0)}\|$  will often be adequate.

**EPS** A REAL\*8 variable which must be set by the user to indicate the accuracy of the solution. The iteration is stopped when the algorithm gives a change  $h_k$  to the approximate solution  $\mathbf{x}_k$  which is smaller than  $\text{EPS} * \|\mathbf{x}_k\|$ . If EPS is chosen too small the routine will stop when the calculation is do-

minated by rounding errors, and EPS will be set to 0.

**MAXFUN** An INTEGER variable. It must be set by the user and its value gives an upper bound for the number of calls of FUNC. If the number of calls of FUNC required exceeds MAXFUN, the routine will return and W(M + 1) will be set to (MAXFUN + 1).

**W** A REAL\*8 array which is used for working space. Its length must be at least  $(mn + 2n^2 + 4m + 12n + 16)$ . On exit it will contain the function values at the solution, i.e.,  $W(j) = f_j(x)$ ,  $j = 1, \dots, M$ . Further W(M + 1) will give the number of calls of FUNC used.

**IW** An INTEGER variable. It must be set by the user to the number of elements in W.

**Input/Output:** None.

**Restrictions:**  $N \geq 1, M \geq 1, N \leq M, DX > 0, DXMAX > 0, STEPL > 0, EPS > 0$ .

**Date of routine:** February 1975.

### Numerical Example

The use of the package is illustrated by a very simple example with two variables: Minimize the objective function  $F(x) = \max_{1 \leq j \leq 3} |f_j(x)|$ , where

$$f_1(x) = 1.5 - x_1(1 - x_2)$$

$$f_2(x) = 2.25 - x_1(1 - x_2^2)$$

$$f_3(x) = 2.625 - x_1(1 - x_2^3).$$

### General Information

**Use of COMMON:** None.

**Workspace:** Provided by the user, see argument W.

This example was given by Beale [4]. For this the user may write the following main program and subroutine FUNC:

```

      IMP_ICIT REAL*8 (A-H,O-Z)
      REAL*8 X(2),W(66)
      EXTERNAL FUNC
      N=2
      M=3
      X(1)=1.
      X(2)=1.
      DX=.2D0
      EPS=1.D-10
      MAXFUN=100
      DXMAX=DX
      STEPL=.001D0
      CALL MINI2W(FUNC,N,M,X,DX,DXMAX,STEPL,EPS,MAXFUN,W,66)
      J=W(M+1)
      WRITE(6,10) (X(I),I=1,N),J,(W(I),I=1,M)
10  FORMAT(' SOLUTION:',F20.14/F30.14/' NUMBER OF CALLS OF FUNC:',14/
1  ' RESIDUALS AT THE SOLUTION:',F20.14/F47.14/F47.14)
      STOP
      END

      SUBROUTINE FUNC(N,M,X,B)
C  BEALE'S TESTFUNCTION
      REAL*8 X(N),B(M)
      B(1)=1.5D0-X(1)*(1.-X(2))
      B(2)=2.25D0-X(1)*(1.-X(2)**2)
      B(3)=2.625D0-X(1)*(1.-X(2)**3)
      RETURN
      END

```

This produces the following results:

```

SOLUTION:      3.00000000000000
               0.50000000000000
NUMBER OF CALLS OF FUNC: 27
RESIDUALS AT THE SOLUTION: -0.00000000000000
                          -0.00000000000000
                          -0.00000000000000.

```

(The program was run on an IBM 370/165 in double precision: 14 hexadecimal digits.)

## REFERENCES

- [1] H. Schjær-Jacobsen and K. Madsen, "Synthesis of nonuniformly spaced arrays using a general nonlinear minimax optimization method," *IEEE Trans. Antennas Propagat.*, vol. AP-24, pp. 501-506, July 1976.
- [2] K. Madsen, "Minimax solution of non-linear equations without calculating derivatives," in *Mathematical Programming Study 3*, M. Balinski and P. Wolfe, Eds. Amsterdam: North-Holland, 1975, pp. 110-126.
- [3] M. J. D. Powell, "The minimax solution of linear equations subject to bounds on the variables," AERE Harwell, OXON, England, Rep. CSS 11, Dec. 1974.
- [4] E. M. L. Beale, "On an iterative method for finding a local minimum of a function of more than one variable," Princeton Univ. Stat. Tech. Res. Group Tech. Rep. 25, 1958.